Guide: “How to report and document computer work in written assignments”

by Bernd Dammann & Allan P. Engsig-Karup

As a part of this course it is mandatory to complete assignments, and these count towards the final grade. Your assignment reports will therefore be used in the overall assessment of your work in the course. Thus, the goal of the assignments should be to clearly inform the reader about the results of a scientific investigation. This implies that, to the extent possible, it should be based on sound theory (if possible), reproducible and verifiable.

The reports do not need to be long or involved, they just need to contain sufficient details to fulfill the requirements of an assignment.

The motivation for requiring reports, is that computers are a great way to generate massive amounts of unorganized data in rather shorts periods of time (in particular for a well-tuned high-performance code base). By requiring that results are organized in a report and synthesized, you as a student are forced to think about what you do, and how to communicate your experiences to others. At the end, this should lead to a greater understanding of the material, (hopefully) catching of errors, identification of opportunities, etc.

It is recommended that the reports contain the points below.   This is **not** a strict layout, but some of the individual points should be covered for each problem/subproblem. Example: if you improve (change) an implementation, describe the changes, report the new results and analyze them. It is completely okay to report “failures” as well, e.g. if a change does not improve the performance as expected – this shows (hopefully) your understanding of the problem/task.

Statement of problem
Which problem is solved, and perhaps in what steps the solution is decomposed into smaller steps (which will then be found in appropriate sections).

Description of hardware and software
Describe clearly what hardware and software was used, and identify what is important to the present study.

Description of algorithms and implementations
What motivates the choices you make? What is used to verify correctness of the implemented code? Which algorithm are you trying to implement? How have you verified your implemented program is doing the right thing?

Experiments and results
Present the important and relevant results of your computations and investigations. Use tables and graphs, and fill these with data that are interesting. For example, you might want to analyze the timing results of your algorithms by providing graphs of the rate of execution (operations/sec) as you vary the size of the problem. You might also want to analyze the rate you achieve compare do the theoretical peak performance rate of the processer to indicate how good the performance is. Make sure to discuss and interpret your results. How do they relate to your expectations? Can we do better? Comment on things that are obviously peculiar.

Conclusions

Discuss your main results and show that you understand the meaning of the results. If your expectations or predictions do not come true, try to explain why. Is it a mistake in the code? Or assumptions?

Your mission should be to convince the reader that you know what the answers should be and that you indeed get them and why this is so. The goal is not to just produce correct solutions – the goal is to demonstrate that you understand what you are doing and that you have been systematic in producing correct results. Thus, this section is very important.

Code listings

Short pieces of relevant code can be placed inside relevant sections of the report, to make it easy to understand what work has been done and how. The complete source code should be handed in as well, e.g. as a ZIP archive that contains all the relevant source files. See the assignment's requirements for details.